AN AUTOMATED CUSTOMER SUPPORT SYSTEM

Field of the Invention

5

This invention relates generally to the field of automated customer support systems, and more particularly pertains to an automated customer support system using intelligent information mining techniques.

10

Background

Support of products is generally a very expensive and time-consuming process. When customers experience problems with a product, the customers typically contact product support for assistance. In such scenarios, efficient query handling and response is essential to provide a quick response that is expected by the customers. Currently

15   such product support services are highly labor-intensive and time-consuming because such customer support services are generally handled by trained personnel who receive the queries from the customers, have to understand the queries, search the product databases, and manually answer the customer's queries. This manual customer support service process can result in wasting human expertise for trivial tasks, which in turn can

20   result in delayed response to the customers and increased cost to a provider of customer support services.

Generally, most of the queries received from customers tend to be repetitious and similar in fundamental characteristics. Such queries can be classified and responded to instantaneously through an automated system. The process becomes more complex

25   when a company has a range of products and a huge customer base having product related problems that has diverse solutions. The process can get even more complex when the media through which the customers interact with customer support services is diverse, i.e., e-mail, internet, telephone, and letter. Further, the process can become even

more complex when response to a query depends on the type of product, the attitude of the customer, and status of the client.

Current processes using automated customer support systems are mostly characterized by huge databases of domain specific keywords, built into the system, 5 using search methods adopted to answer queries specifically through e-mails, and there has been very little effort to build an integrated system to account for various products, customer bases, diverse customer interactions, and the nature of the queries.

Therefore, there is a need in the art for an automated customer support system that can take into account various products, various customer bases, diverse customer 10 interactions, and nature of the queries and respond automatically to customer queries at a low cost as expected by customers.

Summary of the Invention

15 The present invention provides a system and a method for providing an automated customer support for a product that includes a web server to receive one or more queries regarding the product from a customer. A FAQ (Frequently Asked Question) extractor coupled to the web server extracts one or more corresponding product FAQs from respective product FAQ databases. A key-word extractor coupled to 20 the web server and FAQ extractor extracts one or more key words from the received queries and the extracted product FAQ's (Frequently Asked Questions). The key-word extractor further transforms the extracted key-words to unique numerical representations so that the transformed unique numerical representations do not result in multiple similar numerical representations, to avoid ambiguous prediction of meaning from the 25 transformed unique numerical representations. An analyzer coupled to the key-word extractor represents the transformed words into respective query and product FAQ vector forms. The analyzer further applies a convolution algorithm to each of the query vector forms, with each of the product FAQ vector forms separately and obtains one or

more appropriate answers to the queries. Then the analyzer automatically communicates the answers to the customer.

Other aspects of the invention will be apparent on reading the following detailed description of the invention and viewing the drawings that form a part thereof.

5

Brief Description of the Drawings

Figure 1 illustrates an overview of one embodiment of a computer implemented system according to the present invention.

Figure 2 illustrates overall operation of the embodiment shown in Figure 1.

10

Detailed Description

This invention offers an intelligent information mining technique to automatically respond to received customer queries regarding a product. The technique takes into account the nature of the queries including parameters such as product

15   category, mindset of the customer/client, topic, type of customer/client, nature of query and other such information helpful in categorizing the received queries and automatically responding to the customer's queries and providing the answers to the customer.

Figure 1 illustrates an overview of one embodiment of a computer-implemented

20   system 100 according to the present invention. A web server 135 is connected to receive one or more queries regarding a product from a customer from various sources 130. For example, the web server 135 can receive queries from sources such as e-mail, Internet, a data base/data warehouse, Computer telephony interface (CTI), and/or voice recognition systems. The computer-implemented system 100 allows customers 110 to send queries

25   via the various sources 130 via their computers 120, via a letter 122, and/or via telephone 124. In some embodiments, the web server 135 translates the received queries into a text in a natural language. The web server 135 can receive the queries in the letter 122 through a manual entry process. The web server 135 can also translate the received

queries through the telephones into text form using a voice recognition system. In some embodiments, the web server 135 can store the received queries in a buffer 137. The system 100 can receive queries can be regarding a product or a service.

The computer-implemented system 100 can include a categorizer 140. The categorizer 140 is connected to the web server 135 and categorizes the received queries based on a type of query. The categorizer 140 categorizes based on categories such as product category, mindset of the customer, topic, type of customer, nature of query, and/or other such categories that can assist in categorizing the email to automatically communicate the answers to the customer's queries. The following example illustrates one example embodiment of categorizing the received query by the categorizer 140:

> *" I want to comment on your product < XYZ> as I have never seen such a < , worse or more useless > software. All the features should be available very easily and you have made it complicated. I < , regret > having < , invested > in such a < lousy > product."*

The categorizer 140 can categorize the above query as follows:

Product
Mind-set of the customer
❸Topic
❹Request
Status of the customer.

The computer-implemented system 100 includes a FAQ extractor 150. The FAQ extractor 150 is connected to the categorizer 140 and the web server 135 and extracts one or more corresponding product FAQs from respective product FAQ databases 160.

The following illustrates one example of answers to product FAQs that the FAQ extractor can extract from the product FAQ databases for the above-illustrated customer query:

> *"1) To check whether the keyboard got stuck, check whether Num Lock and Caps Lock change their state. If not then the keyboard is hung."*

> *"2) Menus don't work properly when used with keyboard because Num Lock may be on. Try turning Num Lock off."*

In some embodiments, the computer-implemented system 100 manually responds to the received customer's query based on the outcome of the categorization. For example, the categorization can yield a mind-set, a product, and/or status (based on customer's investment in the product) where a manual response may be warranted. In

5   such situations, the computer-implemented system signals a manual response to the customer's queries. In the above-illustrated example, the customer's mind-set is negative, and the computer-implemented system can be programmed to respond manually.

The computer-implemented system 100 includes a key-word extractor 170. The

10   key-word extractor 170 is connected to the web server 135 and the FAQ-extractor 150. The key-word extractor 170 extracts one or more key-words from the received queries and the extracted product FAQs. In some embodiments, the key-word extractor 170 extracts one or more key-words from the received queries and extracted product FAQs based on specific criteria, such as removing one or more general words (such as, *check,*

15   *what, whether, their, then, when, with, because etc.,*) removing all words including three or fewer letters (such as *to, the, and, it, is, not, now, can, you, to, do, off, may, be, on, etc.,*), and/or removing one or more rarely used words. In some embodiments, the system 100 can include a key-phrase extractor to extract one or more key-phrases.

In some embodiments, the key-word extractor 170 replaces the extracted key-

20   words with appropriate synonyms, and further stems the replaced key-words using a Modified Porters Stemming Algorithm.

The following examples illustrate the process of replacing extracted key-words with synonyms:

   computer ← Machine

25   *computer ← System*

The following examples illustrate the process of stemming using the Modified Porters Stemming Algorithm:

   Change → chang

Menus ➔ *menu* ⋅

Properly ➔ *prop*

In some embodiments, the key-word extractor applies the Modified Porters Stemming Algorithm to each of the replaced words until no more stemming can be performed (iterative stemming) on the replaced key-word. This is illustrated in the following examples:

Competitiveness ➔ *competitive* ➔ *compete*

Repeatedly ➔ *repeated* ➔ *repeat*

Further, the key-word extractor 170 transforms each of the extracted key-words into a unique numerical representation so that the transformed unique numerical representations do not result in multiple similar numerical representations, to avoid ambiguous prediction of meaning from the transformed unique numerical representations. The key-word extractor 170 uses a unique text-to-number converter algorithm to transform the extracted key words into unique numerical representations. The following example illustrates the transformed unique numerical representations:

Key-Words	Unique numerical representations
KEYBOARD	*197374*
STUCK	*347959*
NUMLOCK	*260616*
CAPSLOCK	*53845*
CHANG	*58182*
STATE	*47516*
HUNG	*155181*
TELL	*355230*
WORK	*414873*

The key-word extractor 170 normalizes each of the transformed unique numerical representations. In some embodiments, the key word extractor 170

normalizes the transformed unique numerical representations based on minimum/ maximum values in the transformed unique numerical representations. The following algorithm illustrates the process of normalizing using the minimum/ maximum values:

When there are n answers $(A_1, A_2, ...A_n)$ in a product FAQ data base and each

5   answer has few key-words, and the query be Z

$$A_i = \{x_{i1}, x_{i2}, \ldots, x_{ik}\}$$

Normalize the values

$$max = max\ (x_{ij}, z_j)$$

$$x_{ij} \leftarrow x_{ij}\ /\ max$$

10   $$z_j \leftarrow z_j\ /\ max$$

Where, x_{ij} = *the jth keyword extracted from the* i^{th} *answer in the product FAQ.*

z_j = *jth keyword of the email query.*

The following example illustrates the process of normalizing for the example illustrated above using the above-illustrated algorithm:

15   $max= max\ (x_{ij}, z_j)$

$max = \{\ KEYBOARD(\ 197374)$

$STUCK(347959),$

$WORK(414873),$

$TELL(355230)...\}$

20   Normalized values for the above examples are:

KEYBOARD	*0.473517*
STUCK	*0.834782*
WORK	*0.995315*
TELL	*0.852226*

25   An analyzer 180 coupled to the key-word extractor 170 represents the normalized values into respective query and product FAQ vector forms. In some embodiments, the analyzer 180 forms the vectors based on a predetermined number of key-words. The predetermined number is generally in the range of about 10 –15 key-

words (normalized values)'for any given query and answers to product FAQs. The analyzer 180 forms vectors as follows:

$<X_{i\ 1}, X_{i2}, \ldots, X_{in}>$ where X_{ij} represents the j th keyword in the i th FAQ.

Assuming k as the maximum number (predetermined number) of keywords extracted for any answer/ query email.

The email query/answer with keywords $< k$, appends 0's in the rest of the positions.

The following illustrates the formation of query and product FAQ vector forms using 10 key-words (normalized values) for the above-illustrated example:

Query Vector Form for the received query from the customer:

< 0.473517 0.834782 0.995315 0.852226 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 >

Product FAQ vector forms for extracted answers from product FAQ:

1) < 0.473517 0.834782 0.625239 0.129179 0.139583 0.833720 0.372292 0.000000 0.000000 0.000000 >

2) <0.557209 0.193915 0.995315 0.704841 0.916632 0.473517 0.092645 0.625239 0.878554 0.530550 >

The analyzer 180 applies a convolution algorithm to each of the query vector forms, with each of the product FAQ vector forms separately and obtains one or more appropriate answers to the queries received from the customer. The convolution algorithm is based on a moving weighted average with time domain representation of the key-words flipped back. The convolution technique involves matching the customer's queries to the answers in the product FAQs to pick the most relevant answers to the queries. The convolution algorithm used by the analyzer 180 to obtain the answers to the customer's queries is as follows:

$$y[m] = \sum_{k=0}^{m} x[k] * h[m-k]$$

Where m varies from 0 to 2n-1

$y[m]$ - the convoluted time domain

representation of the keyword

5    $x[k]$ – the original time domain

representation of the keyword

$h[m-k]$ – flipped and shifted time domain

representation of the keyword

In some embodiments, the analyzer 180 applies self-convolution algorithm to

10   each of the query vector forms to obtain resultant convoluted representation of the

query. The self-convolution algorithm used by the analyzer 180 is as follows:

$$Z[m] = \sum_{k=0}^{m} x[k] * x[m-k]$$

Where m varies from 0 to 2n-1

$Z[m]$ - the self-convoluted vector for the e-mail query

15   $x[k]$ - the original vector corresponding to the query

$x[m-k]$ - the flipped vector

The following example illustrates the application of the above self-convolution

algorithm tot he above-illustrated query vector form:

20   x [k] = < 0.473517   0.834782   0.995315   0.852226   0.000000
                  0.000000   0.000000   0.000000 0.000000 0.000000 >

$$Z[0] = \sum_{k=0}^{0} x[k] * x[0-k]$$ = x[0] * x[0]   = 0.473517 * 0.473517   = 0.224218

$$Z[1] = \sum_{k=0}^{1} x[k] * x[1-k]$$ = x[0] * x[1] + x[1] * x[0]   = 0.473517 * 0.834782 +0.834782* 0.473517   = 0.790567

25   $$Z[2] = \sum_{k=0}^{2} x[k] * x[2-k]$$ = x[0] * x[2] + x[1] * x[1] +x[2] * x[0]

= 0.473517 * 0.995315 + 0.834782 * 0.834782 + 0.995315 * 0.473517   = 1.639458

$$Z[6] = \sum_{k=0}^{6} x[k] * x[7-k] = \text{x[0] * x[6] + x[1] * x[5] +x[2] * x[4] + x[3] * x[3] + x[4] * x[2] +x[5] * x[1] +x[6] * x[0]}$$

$$= 0.473517 * 0 + 0.834782 * 0 + \ldots \qquad + 0 * 0.473517 = 0.726289$$

Similarly, application of the above algorithm to $z[7], z[8], \ldots z[19]$ yields *0.000000*.

Based on the above illustrated self-convolution computation, the algorithm

yields the following resultant convoluted representation of the query:

0.224218 0.790567 1.639458 2.468829 2.413498 1.696467 0.726289 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.0000000

Further, the analyzer 180 applies the convolution algorithm to the query vector

form, with each of the product FAQ vector forms separately to obtain resultant

convoluted representation of the answer using the following convolution algorithm:

$$y_i[m] = \sum_{k=0}^{m} x[k] * h[m-k]$$

Where m varies from 0 to 2n-1

$y_i [m]$ - the convoluted time domain representation for i[th] answer vector .

$x[k]$– the original time domain representation corresponding to the email query

vector.

$h[m-k]$ – flipped time domain representation corresponding to the answer vector

in the product FAQ.

The following are the resultant convoluted representation of the FAQ answers

obtained by using the above convolution algorithm to the above-illustrated two product

FAQ vector forms:

0.224218 0.790567 1.464221 1.817521 1.507664 1.172721 1.12128
1.259553 1.081066 0.317277 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000

0.27171 0.750754 1.16439 1.653343 1.893364 2.19763 2.288878
1.553774 0.600691 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000   0.000000 0.000000 0.000000   0.000000

Then the analyzer 180 compares the above resultant convoluted representation of the query and the FAQ answers, to determine one or more appropriate answers to the queries. In some embodiments, the analyzer 180 compares each of the resultant convoluted representation of the query with each of the resultant convoluted representation of the FAQ answer to find the answers to the customer's queries. The analyzer uses the following algorithm to find the relevant answers:

If y_i[j]-z[j]= 0 then

count[i]=count[i]+1;   j =0 to 2n −1, i = 0 to n-1

In some embodiments, the matching of the resultant convoluted representations is accomplished by using the following algorithm:

$$Error(i) = \sum_{j=0}^{2n-1} \sqrt{((y_i[j] - z[j]) * (y_i[j] - z[j]))}$$

Min error = min (*Error (i)*) $\forall$ i = 0 to n-1

Minimum error among the sum of difference between the resultant convoluted representation of the query and the FAQ's, computed at each of the peaks can be the answer for the customer's query.

The analyzer 180 automatically communicates the answers to the customer's queries using various communicating methods 190. For example, the analyzer 180 can automatically communicate the answers using communicating methods, such as e-mails, telephones, and/or letters. The analyzer 180 can use text to voice/speech conversion for phone communications, and text to manual mail conversion for sending letters to the customers.

Figure 2 illustrates an overview of one embodiment of the process 200 of the present invention. This process 200 provides, as illustrated in element 210, a computer-implemented system including a web server. The web server receives one or more

queries regarding a product from a customer. Some embodiments allow the web server to receive queries related to multiple products and/or services from the customer. The web server can receive the queries from the customer from various sources such as a letter, a telephone, e-mail and Internet. Some embodiments allow the text to be

5 translated to any natural language.

Element 220 includes storing the received queries. Some embodiments allow the text in a received letter to be manually entered into a buffer. Some embodiments allow the queries received from a fixed/mobile telephone to be converted into text using a voice recognition system and further store the converted text in the buffer.

10 Element 230 includes categorizing the received/stored queries based on a type of query. Categorizing the queries based on a type of query can include categorizing based on various categories such as product, mindset of customer, topic of the query, type of customer, nature of query, and/or other such categories that can assist in categorizing the received queries.

15 Element 240 requires checking the received queries to see whether they require a manual response. This can be based on the outcome of the categorization of the received queries. For example, if the customer query suggests an angered customer or the customer is a highly valued customer to the business, then the system can request a manual response as indicated in Element 245 to the customer to satisfy the customer's

20 needs in these special situations.

Element 250 includes one or more answers from product FAQ's related to the received queries from a product FAQ database. Further, element 250 includes extracting query and FAQ key words from the received queries and extracted answers, respectively. Some embodiments allow extracting key words based on specific criteria

25 such as removing one or more general words from the queries and product FAQs, removing all words including three or fewer letters, and removing one or more rarely used words. Some embodiments allow the extracted key words to be replaced by appropriate synonyms. Further, some embodiments allow stemming the replaced key

words using a Modified Porters Stemming Algorithm. In some embodiments, the Modified Porters Stemming Algorithm is applied to the stemmed words until no more stemming can be performed. The process of stemming is described in detail with reference to Figure 1.

5   Element 260 includes transforming each of the extracted query and FAQ key words into unique numerical representations such that the transformed unique numerical representation does not result in multiple similar numerical representations, to avoid ambiguous prediction of meaning of the translated words in the received queries and extracted answers. In some embodiments, the transformed key words (unique numerical

10   representations) are normalized. Some embodiments allow normalizing based on minimum/ maximum values in the transformed unique numerical representations. The process of normalizing is described in detail with reference to Figure 1.

Element 270 includes representing the transformed query and FAQ key words in query and product FAQ vector forms, respectively. The process of representing the

15   vector forms is discussed in more detail with reference to Figure 1. Some embodiments allow representing the vector forms using the normalized unique numerical representations. In some embodiments, the size of the vector forms is based on a predetermined number of transformed key words. The size of the predetermined number of key words can be in the range of about 10 to 15 key words.

20   Element 280 includes applying a convolution algorithm to each of the query vector forms, with each of the product FAQ vector forms separately and obtains one or more appropriate answers to the customer's queries. In some embodiments, Element 280 includes applying a self-convolution algorithm to each of the query vector forms to obtain resultant convoluted representation of the query. In addition, Element 280 can

25   also include applying the convolution algorithm to the query vector form with each of the product FAQ vector forms to obtain resultant convoluted representation of the FAQ answers. Further, Element 280 includes comparing the resultant convoluted representation of the query and the FAQ answers to determine appropriate answers to

the queries. The process of applying convolution and self-convolution algorithms is described in detail with reference to Figure 1.

Element 290 includes automatically communicating the answers to the queries received from the customer. Some embodiments allow the answers to be communicated

5   to the customers using communication methods such as a letter, an email, Internet and a telephone.

The above description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those skilled in the art. The scope of the invention should therefore be determined by the appended claims, along with the full

10   scope of equivalents to which such claims are entitled.